

Pixel Matrix Shield Bedienungsanleitung Keystudio WS2812

Sehr geehrter Kunde,

vielen Dank, dass Sie sich für unser Produkt entschieden haben.

Im Folgenden haben wir aufgelistet, was bei der Inbetriebnahme zu beachten ist:

Schritt 1 – Anschließen des Shields



Bild 1: Keystudio Matrix Shield WS2812

Stecken Sie die LED-Matrix so auf den Arduino auf, so dass er in sämtlichen PINs der Arduino PIN-Steckerleiste aufgesteckt wird.

Eine weitere Verkabelung ist nicht notwendig, da diese LED-Matrix über den Arduino mit Strom versorgt wird.

Schritt 2 – Installation der Bibliotheken

Nachfolgend können Sie ein funktionsfähiges Codebeispiel entnehmen und auf Ihren Arduino übertragen.

Unterschiedliche Beispielbeleuchtungen, wie z.B. das nacheinander ablaufende Auffüllen der LEDs mit einer Farbe, sind bereits vorhanden.

Die Farben werden in RGB mit Zahlen von 0 bis 255 (z.B. (255, 0, 0) für rot) bezeichnet.

Bereits vorhandene Farbläufe sind die Funktionen **colorWipe**, **theaterChase**, **rainbow**, **rainbowCycle**, **theaterChaseRainbow**.

Diese können, nach eigenem Belieben, innerhalb der **void loop()** entfernt oder vervielfältigt werden.

Bevor Sie jeglichen Code auf den Arduino übertragen, muss die **Adafruit NeoPixel** Bibliothek hinzugefügt werden.

Dazu klicken Sie bitte, wie in Bild 2 zu sehen, auf Sketch → Bibliothek einbinden → Bibliotheken verwalten.

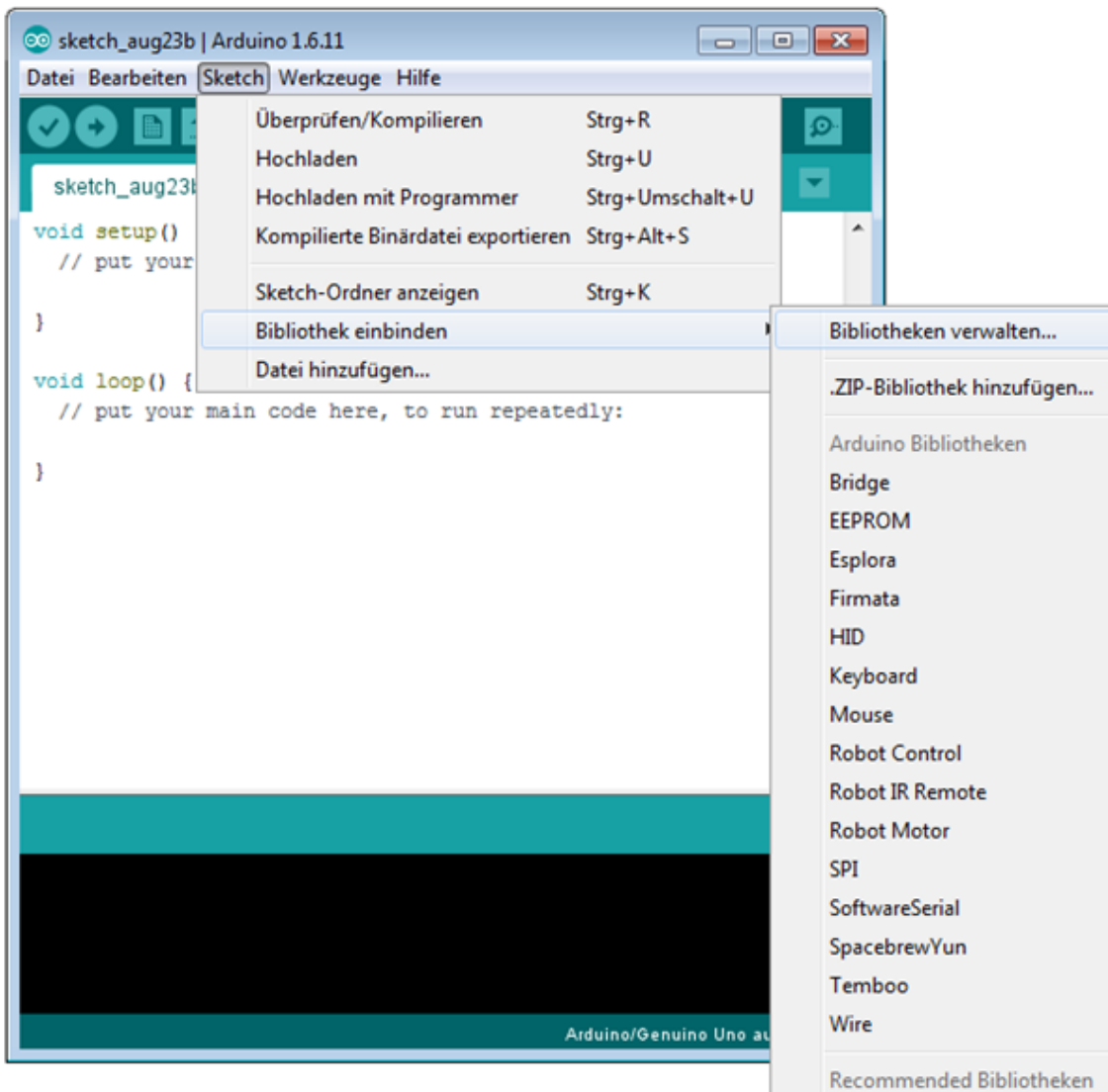


Bild 2: Bibliothekverwaltung im Arduino-Sketch

Im sich darauf öffnenden Bibliotheksverwalter suchen Sie unter dem Suchbegriff „**Neopixel**“ die **Adafruit Neopixel** Bibliothek und installieren diese.

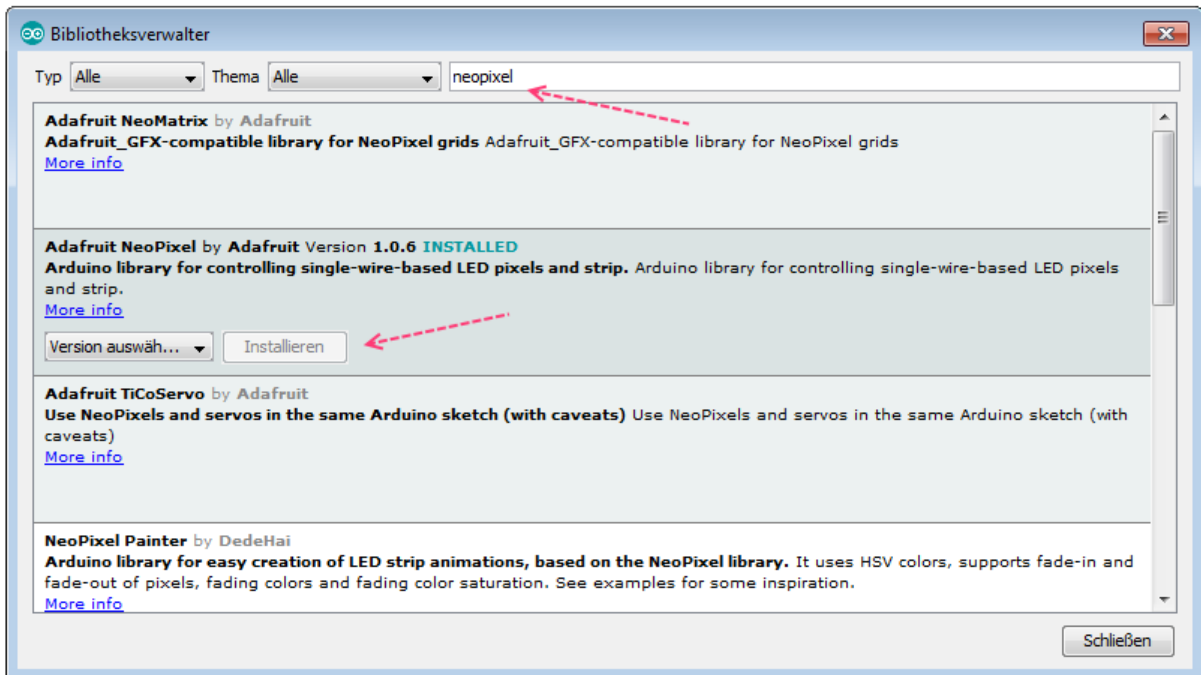


Bild 3: Installation der Adafruit NeoPixel Bibliothek

Schritt 3 – Installation der Matrix

Nach der Installation der Bibliothek kann das nachfolgende Codebeispiel übernommen und auf den Arduino übertragen werden.

```
#include <Adafruit_NeoPixel.h>
#ifdef __AVR__
#include <avr/power.h>
#endif
#define PIN 13
//Grundeinstellung (Anzahl der LEDs, usw.)
Adafruit_NeoPixel strip = Adafruit_NeoPixel(40, PIN, NEO_GRB + NEO_KHZ800);
void setup()
{
  strip.begin();
  strip.show(); // Alle LEDs initialisieren
}
```

Code 1: Teil 1 der LED-Matrix

```

void loop() {
// Beispielhafte Farbläufe:
farbDurchlauf(strip.Color(255, 0, 0), 50); // Rot
farbDurchlauf(strip.Color(0, 255, 0), 50); // Grün
farbDurchlauf(strip.Color(0, 0, 255), 50); // Blau
// Theater-Beleuchtung:
theaterBeleuchtung(strip.Color(127, 127, 127), 50); // Weiß
theaterBeleuchtung(strip.Color(127, 0, 0), 50); // Rot
theaterBeleuchtung(strip.Color(0, 0, 127), 50); // Blau
regenbogen(20);
regenbogenVerlauf(20);
theaterRegenbogen(50);
}
// LEDs nacheinander mit angegebener Farbe auffüllen
void farbDurchlauf(uint32_t c, uint8_t wait) {
for(uint16_t i=0; i<strip.numPixels(); i++) {
strip.setPixelColor(i, c);
strip.show();
delay(wait);
}
}
// Regenbogen
void regenbogen(uint8_t wait) {
uint16_t i, j;
for(j=0; j<256; j++) {
for(i=0; i<strip.numPixels(); i++) {
strip.setPixelColor(i, Wheel(((i+j) & 255)));
}
strip.show();
delay(wait);
}
}
// Leicht veränderter Regenbogen
void regenbogenVerlauf(uint8_t wait) {
uint16_t i, j;
for(j=0; j<256*5; j++) {
for(i=0; i< strip.numPixels(); i++) {
strip.setPixelColor(i, Wheel(((i * 256 / strip.numPixels()) + j) & 255));
}
strip.show();
delay(wait);
}
}
}

```

Code 2: Teil 2 der LED-Matrix

```

// Theater-Beleuchtung
void theaterBeleuchtung(uint32_t c, uint8_t wait) {
for (int j=0; j<10; j++) { //10 Wiederholungen
for (int q=0; q < 3; q++) {
for (int i=0; i < strip.numPixels(); i=i+3) {
strip.setPixelColor(i+q, c); //Aktiviere jede 3. LED
}
strip.show();
delay(wait);
for (int i=0; i < strip.numPixels(); i=i+3) {
strip.setPixelColor(i+q, 0); //Deaktiviere jede 3. LED
}}}}

// Theater-Beleuchtung mit Regenbogen-Effekt
void theaterRegenbogen(uint8_t wait) {
for (int j=0; j < 256; j++) { // Alle 256 Farben durchlaufen
for (int q=0; q < 3; q++) {
for (int i=0; i < strip.numPixels(); i=i+3) {
strip.setPixelColor(i+q, Wheel( (i+j) % 255)); //Jede 3. LED aktivieren
}
strip.show();
delay(wait);
for (int i=0; i < strip.numPixels(); i=i+3) {
strip.setPixelColor(i+q, 0); //Jede 3. LED deaktivieren
}}}}

uint32_t Wheel(byte WheelPos) {
WheelPos = 255 - WheelPos;
if(WheelPos < 85) {
return strip.Color(255 - WheelPos * 3, 0, WheelPos * 3);
}
if(WheelPos < 170) {
WheelPos -= 85;
return strip.Color(0, WheelPos * 3, 255 - WheelPos * 3);
}
WheelPos -= 170;
return strip.Color(WheelPos * 3, 255 - WheelPos * 3, 0);
}

```

Code 3: Teil 3 der LED-Matrix